

Abstract

This project includes the imagining, design, build, and test of a web application that creates and tracks a user's progress on completing tasks that an administrator has created for the user. The goal of this project is to have a functioning webpage that is robust and scalable to support many users and many tasks. The application will be developed for use on all modern web browsers, and will have a persistent server to access from any platform. This project was designed to be an exercise in building a modern web application, and as such is written using many different languages, APIs and libraries. The front-end is setup in HTML and CSS, with an approach that a web-designer might update the aesthetic of the application. MySQL is utilized for the server databasing implementation, and PHP, JavaScript and JQuery are used as the bridge between the database and the front-end of the application.

Introduction

For my senior project I wanted to explore the world of web-development. I have always been intrigued by the internet, and it excited me to approach the web as a creator rather than a user. I found that the best way for me to learn more about web-development was to create a project similar to those I might be working on in industry. Therefore I approached this project in a way that is relevant to the contemporary web design paradigm. This includes server-side databasing, a straightforward login scheme, easy-to-use administrator tools, as well as a simple and approachable user interface. Similar to the design approach of this project, the implementation has been chosen specifically; it is written using the most current and popular web languages.

The application idea originally stemmed from creating a framework for competition in a club or group of people. A club or group administrator can add tasks or "badges" which users can complete. Users will be able to track their progress when they log into their account, which will display the badges they have completed, and the badges that remain unfinished.

Background

Relevancy and Standards

This project is developed using a standard, current, and popular web development paradigm. As stated previously, the goal of this project was a deeper understanding of the contemporary web development world, and therefore this application was developed with today's standard in mind. This standard includes a server-side relational database (MySQL),

server-side database management (PHP), interconnectivity between the webpage and the database (AJAX, JQuery), browser-side data control and HTML generation (JavaScript), web display and layout (HTML, CSS). I chose this approach because it utilizes the power and flexibility of a relational database, as well as having limitless access as it is an internet-based application.

Research

When deciding how to implement this project I entertained a few design options, I found a few approaches that I had not previously worked with, and through my research I was able to gain a more complete understanding of the options when creating a web application.

One noteworthy coding language that I learned about through this project was Node.js. Node.js is still relatively new, as it was initially released in 2009, but it is quickly becoming more popular. I attempted to find a way to design my application using Node.js but was unable to due to how Node.js handles database connectivity. Despite putting much effort into finding a solution to connect Node.js to my server-side MySQL database I was unable to do so due to lack of this kind of support with Node.js. Although it is unfortunate that I was unable to develop my project in Node.js I am happy I was able to explore it and will certainly utilize what I learned in a future project.

Scope

The scope of this project is most intimately related to enhancing my skill and understanding of web development. Cal Poly has helped me become a better programmer, but most of the material has been low-level, such as C programs or embedded systems work. However, I was able to take three relevant technical electives that gave me a base to stand on when approaching this project. Notwithstanding, this project required me to teach myself as I went.

The technical electives that I took were intro/advanced JavaScript and introduction to databases. Through these classes I learned the fundamentals of JavaScript and MySQL, such as how the Document Object Model works and how a relational database is maintained and accessed. Although these classes were important to this project, the majority of the application required me to find out what I needed to create my app, and how to use the tools that I found.

Existing Solutions

OpenBadges is a website that is very similar to my idea, it is a website that tracks an individual's progress on completing "badges" which they have outlined on their website. This website was created and is maintained by Mozilla and has many more features than my application. These features include but are not limited to: graphical representation of badges, sequential unlocking of badges, user submitted badges and meta-progress tracking. I found this website after I began developing my application, and it has served me as an inspiration as to

what a professional design team can accomplish using the same tools that I have in creating their website. OpenBadges was developed using similar design choices to this application.

Materials

Physical materials for this project are limited to my laptop (Macbook Pro) as this is exclusively a software project.

Objectives

The objective of this project is to have a functional progress tracking application. Including the ability for an administrator to log in and create badges for users to complete, and enabling users to log into their account so that they might view the badges posted for them to complete. The user interface should be simple and straightforward where the user can clearly see their to-do list and also look at the badges they have already completed. Users will be able to login from anywhere and using any web browser.

Marketing & Engineering Requirements

- Basic functionality of the application requires the ability of creation and deletion of user accounts.
 - Engineering Requirement: An administrator account is required to monitor other users of the application. This administrator must have privileges to add and delete users.
- Basic functionality also requires the ability of creation and deletion of badges.
 - Engineering Requirement: The administrator account must be able to add and remove badges from the badge database.
- This application works on all operating systems.
 - Engineering Requirement: The programming languages utilized for this application must be operating system independent, and fully functional on all operating systems.
- This application works on all web browsers.
 - Engineering Requirement: The project must be designed with all contemporary web browsers in mind. There are subtle differences when designing for different web browsers especially with HTML & CSS, and those must be accounted for.
- This application is scalable.
 - Engineering Requirement: The application must be designed to accept many users and badges, scaling appropriately to accept these new users and badges.
- The application can be improved.
 - Engineering Requirements: The design of this application must have further expansion in mind. The application must be scalable not only for the amount of

users and badges it may accept, but also the amount of new features that future programmers might add.

Development Process Workflow

This project was developed during fall quarter of 2014, after deciding on the project idea, development followed a few phases that took approximately a week each. The phases are as follows:

Phase 0: Researching different programming languages and design decisions that will impact the overall construction of the program. Selecting which languages to use and how they will all interact together to achieve a working product.

Phase 1: Planning a logical workflow that will enable me to finish the program before it is due. As described below, my workflow is very modular, emphasising working on many small parts of the program first and later combining them into the application.

Phase 2: Writing a text-based JavaScript program which simulates overall program flow. Creating this text-based version of the program illuminated many new design issues and concerns that I addressed as I continued the project.

Phase 3: Creation of the relational database for the program in MySQL. This includes designing and implementing the MySQL framework which my database will follow.

Phase 4: Writing the PHP code that will manage the server and accept AJAX calls from the client.

Phase 5: Writing the AJAX calls that will dynamically send requests to the server based on the current user and data desired to be returned or updated.

Phase 6: Creating the HTML skeleton that includes static information to be displayed as well as creating the framework which will be used to accept dynamic HTML blocks generated in JavaScript.

Phase 7: Combining the above phases into one working application that has a front-end to interact with, and that communicates with the server appropriately.

Phase 8: Tweaking the front-end of the application with CSS and JQueryUI so that it is more user-friendly and approachable.

Phase 9: Testing and quality control of the application.

Project Breakdown / Languages Utilized

HTML (Hypertext Markup Language): This application utilizes HTML as a skeleton and to display information, and it is designed for any web-browser. HTML is fundamental to the internet as it has been for many years. My application incorporates some static HTML: such as the login page, and also includes dynamic HTML generated on page load by JavaScript such as the completed and uncompleted badge lists.

CSS (Cascading Style Sheets): I utilize CSS in this application to change the aesthetic of the application. CSS is used specifically for spacing different elements on the page, changing fonts, changing colors, layout, and overall aesthetic. These aspects can be changed in the source file `Badge.css` which will update the aesthetic throughout the application.

JS (JavaScript): JavaScript is used to dynamically send requests via AJAX to the server so that it can post the responses to the HTML skeleton. JavaScript also allows me to utilize JQuery and JQueryUI.

JQuery: JQuery enables me to quickly update the DOM (Document Object Model) with the data that has been retrieved from my database.

JQueryUI: I use JQueryUI to improve the aesthetic of the application. Different styles and colors can be selected and implemented into this application easily due to how JQueryUI is setup and integrated into the application. One example use of JQueryUI in this application is the accordion widget utilized in the user-home page.

AJAX (Asynchronous JavaScript and XML): I use the JQuery `.ajax()` command to create a dynamic request based on the current user to run PHP code that will either update or fetch data from the database depending on the context of the `.ajax()` call. This is what connects the web front end to the PHP and server portions of this application.

PHP (Hypertext Preprocessor): PHP is utilized in this application when the server receives an AJAX request from the client. The PHP code takes the AJAX request, and based on the parameters sent to the server, will dynamically generate a MySQL statement that will either update or get information from the database.

MySQL (My Structured Query Language): MySQL is a relational database language used in this application to create and maintain the user and badge records. The MySQL requests are generated dynamically and executed from PHP code, as is described above.

Diagrams & Descriptions

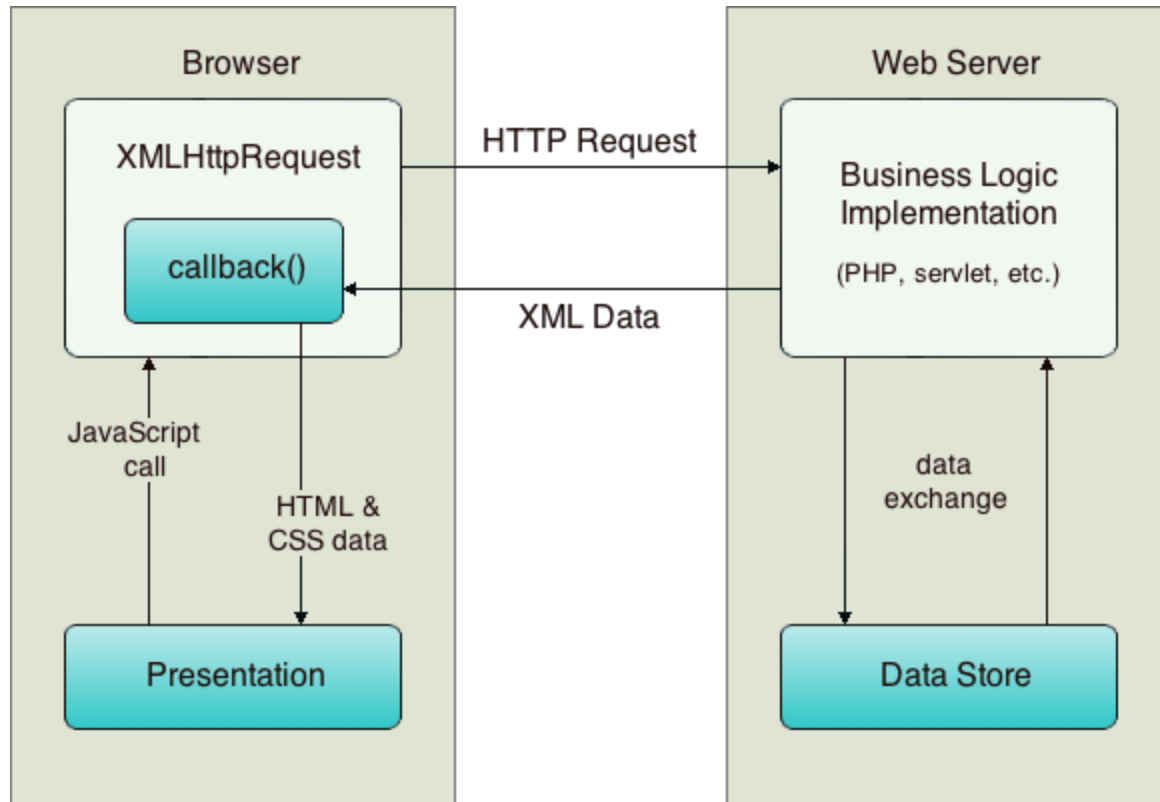


Figure 1. (<http://tinyurl.com/l5l63sz>)

Figure 1 describes the overall application flow. On the left side of the diagram, the browser handles the presentation, which in turn uses Javascript to make a AJAX request to the web server. When the web server receives the AJAX request, it uses PHP to process the request, and access the datastore(MySQL server) which is then sent back to be processed by PHP and send back as XML data to the browser. Once the browser receives its response, it uses the appropriate callback function to generate the HTML & CSS so that it can add it to the presentation.

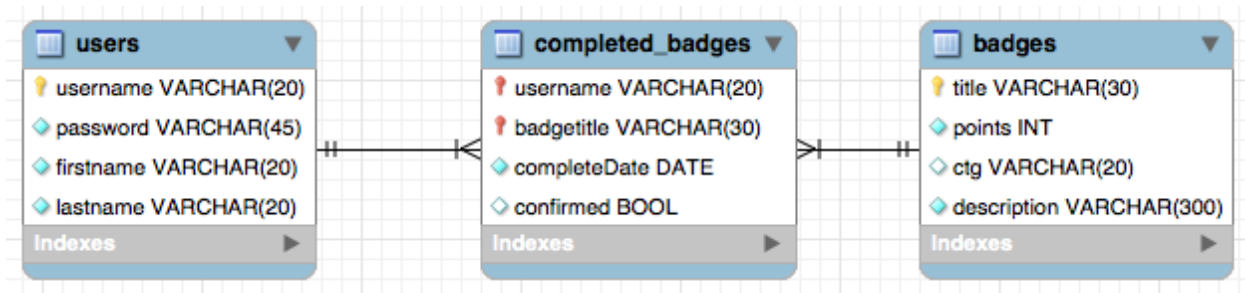


Figure 2.

Figure 2 outlines the MySQL database for the application, BadgeDB. BadgeDB includes three tables, users, badges and completed_badges. The columns in each table are described in the diagram, as well as an indication on how the tables are connected. The users to completed_badges and badges to completed_badges connection are both one-to-many type connections, where a user might have many completed badges and a badge may be completed by many users. When a user completes a badge, an entry is made into the completed_badges table to record that completion. Checking the completed_badges table for a specific user and badge will indicate if a user has completed a specific badge or not.

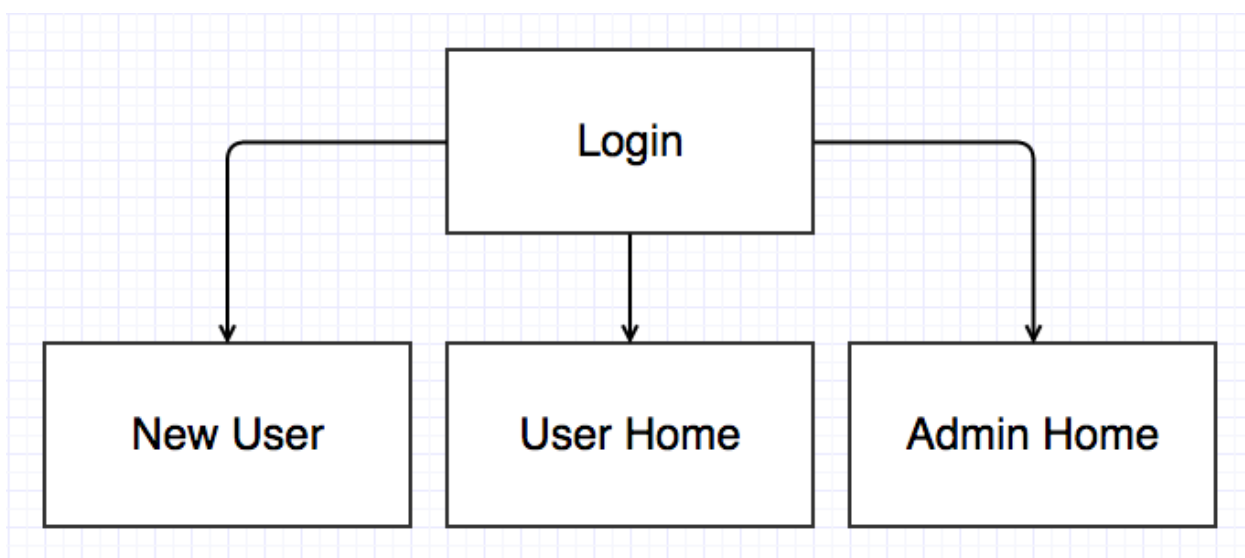
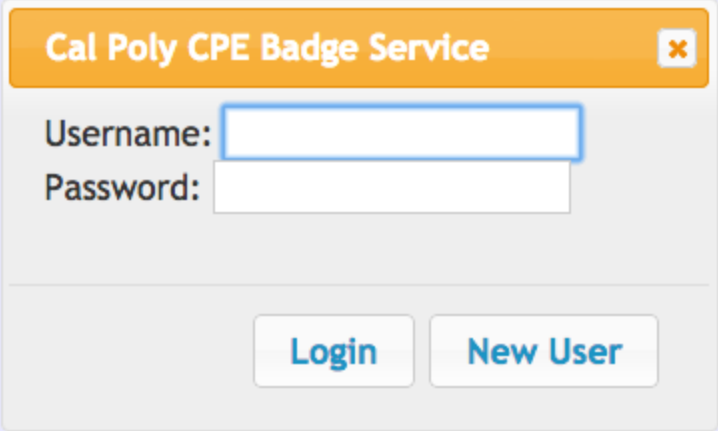


Figure 3.

Figure 3 illustrates the program use flow. This application has three main pages, the login, user home, and administrator home pages. The power of this application is highlighted by the simplicity of the use flow; as despite the fact that the program is simple for every user, each user will have a different home page that is tied to their account. This design is widely used in web-development as it is not only straightforward for the user, but also gives the developer an easy way to deploy new features for all users. New users start by going to the New User page, to create an account and then use the application by logging in accessing their User Home page. The administrator page can be accessed with the account “admin” and password “cpadmin”.

The following screenshots illustrates the pages outlined in Figure 3.



A screenshot of a login window titled "Cal Poly CPE Badge Service" with a close button (X) in the top right corner. The window has a light gray background. It contains two input fields: "Username:" and "Password:". Below these fields are two buttons: "Login" and "New User". The "Username:" field is highlighted with a blue border.

Figure 4. Login Screen

Signup!

Username:

Password:

First Name:

Last Name:

Figure 5. New User Screen

Welcome rgreen02

Unfinished Badges:

▶ [Football Game](#)

▼ [Office Hours](#)

Points: 10

Category: Meetings

Description: Go to OH

[Complete Badge](#)

Completed Badges:

▶ [CPE Office](#)

▶ [Register at the Gym](#)

▶ [Tech Talk](#)

▼ [Unix Servers](#)

Points: 10

Category: CSC Introduction

Description: Log into one of the CSC Unix servers!

Figure 6. User Home Screen

Administrator Home

Add Badge

Badge Title:

Point Value:

Category:

Description:

Create new Badge

Badge List:

▶ CPE Office	▶ andrew1
▶ Football Game	▶ guy1
▶ Office Hours	▶ joliver
▶ Register at the Gym	▶ laurylcreative
▶ Tech Talk	▶ rgreen02
▼ Unix Servers	▼ warren

Points: 10
Category: CSC Introduction
Description: Log into one of the CSC Unix servers!

Delete Badge

User List:

Password: warren
First Name: Warren
Last Name: Warren

Delete User

Figure 7. Administrator Home Screen

Future Improvements / Iterations

This application has many areas for expansion and improvement, and was designed with these expansions in mind. Possible things to add or improve are as follows:

Improved Aesthetic: The aesthetic of this app could be improved with more attention to color and positioning, as well as possibly using a different UI library then JQueryUI, such as Bootstrap.

Picture Based Badges: An upgrade to the aesthetic of this app might be changing the badges from being text based, to having a logo or picture for each badge.

Improved Database: The database currently works correctly, but it could be expanded on to add more relevant data to both users, badges, and completed_badges. This information could be used to add more features such as tiered badges.

Tiered Badges: Tiered badges, that could be unlocked additively could be implemented so that users could complete many small badges that unlock one large badge.

Grouped Badges: Utilizing the category(ctg) tuple could lead to a straightforward way to implement badge grouping, which could give a user a better description of how many badges of a specific type they had completed.

Grouped Users: Grouping users could add for a team dynamic among people who are using the same application.

Badge Suggestion: Adding a badge suggestion tool would allow users to submit to the admin user a badge that they think might be a good addition to the application.

Concluding Thoughts & Acknowledgements

This project has been a valuable learning experience for me and has helped me build on both my web-development and overall programming ability. Working on this project has also increased my confidence as a programmer and developer as I was forced to imagine, design, and motivate myself throughout the development process. I started this project without much experience in web-development and now looking back on how much I have learned, I know that this was time well spent. I intend to continue working on this project over the next year, adding both aesthetic polish, and more features. Eventually I plan on adding this project to my resume as a showcase of my ability. Creating this website has given me the confidence and ability to be able to work for a team like Mozilla and help add to a page such as OpenBadges. Special thanks to Dr. John Oliver, my project advisor, for his time and support; our weekly meetings helped me stay focused and on-time.

Appendix: Source Files

MySQL Database Initialization

```
CREATE TABLE IF NOT EXISTS `users` (  
  `username` varchar(20) NOT NULL,  
  `password` varchar(45) NOT NULL,  
  `firstname` varchar(20) NOT NULL,  
  `lastname` varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `badges` (  
  `title` varchar(30) NOT NULL,  
  `points` int(11) NOT NULL,  
  `ctg` varchar(20) DEFAULT NULL,  
  `description` varchar(300) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `completed_badges` (  
  `username` varchar(20) NOT NULL,  
  `badgetitle` varchar(30) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Admin.js

```
var admin = function() {  
  ;  
}
```

```
admin.prototype.configure = function() {  
  var self = this;  
  var BadgeList = $('#BadgeList');  
  var UserList = $('#UserList');  
  var scriptString = 'this string';  
  var myData;
```

```
  $('#BadgeList').accordion({collapsible: true});  
  for (var i = 0; $('#titleEntry' + i).length; i++) {  
    BadgeList.append($('#titleEntry' + i))  
      .append($('#divEntry' + i))  
      .append($('#pointsEntry' + i))  
      .append($('#categoryEntry' + i))  
      .append($('#descriptionEntry' + i))  
      .append($('#<button id="deleteBtn' + i + '" title="' + $('#titleEntry' + i).text() + '>Delete Badge</button>''));
```

```
  $('#deleteBtn' + i).button().on('click', function(e) {  
    myData = 'content_txt=' + $(this).attr('title');  
    var request = $.ajax({  
      url: "delete_badge.php",  
      type: "POST",  
      data: myData,  
      dataType: "html"  
    });
```

```
    request.done(function(msg) {  
      //console.log(msg);  
      location.reload();  
    });
```

```

        request.fail(function(jqXHR, textStatus) {
            alert( "Request failed: " + textStatus );
        });
    });
    console.log($('#titleEntry' + i).text());
}
BadgeList.accordion("refresh");

$('#UserList').accordion({collapsible: true});
for (var i = 0; $('#utitleEntry' + i).length; i++) {
    UserList.append($('#utitleEntry' + i))
        .append($('#udivEntry' + i)
        .append($('#passwordEntry' + i))
        .append($('#firstnameEntry' + i))
        .append($('#lastnameEntry' + i))
        .append($('#<button id="udeleteBtn' + i + '" title="' + $('#utitleEntry' + i).text() + '">Delete User</button>')));

    $('#udeleteBtn' + i).button().on('click', function(e) {
        myData = 'content_txt=' + $(this).attr('title');
        var request = $.ajax({
            url: "delete_user.php",
            type: "POST",
            data: myData,
            dataType: "html"
        });

        request.done(function(msg) {
            //console.log(msg);
            location.reload();
        });

        request.fail(function(jqXHR, textStatus) {
            alert( "Request failed: " + textStatus );
        });
    });
    console.log($('#utitleEntry' + i).text());
}
UserList.accordion("refresh");
$('#createBtn').button();
}

```

Admin.php

```

<?php
$username = "root";
$password = "root";
$hostname = "localhost";

if(!isset($_COOKIE['adminloggedin'])) {
    header("location:index.php");
}

$dbhandle = mysql_connect($hostname, $username, $password)
or die("Could not connect to database");

$selectd = mysql_select_db("badgedb", $dbhandle);

```

```

if(isset($_POST['title']) && isset($_POST['points'])
&& isset($_POST['ctg']) && isset($_POST['desc'])) {

    $title = $_POST['title'];
    $points = $_POST['points'];
    $ctg = $_POST['ctg'];
    $desc = $_POST['desc'];

    $query = mysql_query("SELECT * FROM badges WHERE title='$title'");
    if(mysql_num_rows($query) > 0 ) { //check if there is already an entry for that username
        echo "A badge with that title already exists!";
    }
    else {
        mysql_query("INSERT INTO badges (title, points, ctg, Description)
        VALUES ('$title', '$points', '$ctg', '$desc')");
        header("location:admin.php");
    }
}
$query = mysql_query("select b.title, b.points, b.ctg, b.description
from badges b");

for ($i = 0; $row = mysql_fetch_array($query, MYSQL_BOTH); $i++) {
    echo "<h3 class='subListEntry' id='titleEntry$i'>$row[0]</h3>";
    echo "<div class='subListEntryExpansion' id='divEntry$i'></div>";
    echo "<li id='pointsEntry$i'>Points: $row[1]</li>";
    echo "<li id='categoryEntry$i'>Category: $row[2]</li>";
    echo "<li id='descriptionEntry$i'>Description: $row[3]</li>";
}

$query = mysql_query("select u.username, u.password, u.firstname, u.lastname
from users u");

for ($i = 0; $row = mysql_fetch_array($query, MYSQL_BOTH); $i++) {
    echo "<h3 class='subListEntry' id='utitleEntry$i'>$row[0]</h3>";
    echo "<div class='subListEntryExpansion' id='udivEntry$i'></div>";
    echo "<li id='passwordEntry$i'>Password: $row[1]</li>";
    echo "<li id='firstnameEntry$i'>First Name: $row[2]</li>";
    echo "<li id='lastnameEntry$i'>Last Name: $row[3]</li>";
}

mysql_close();
?>

<html>
<head>

<link href="jquery/css/ui-lightness/jquery-ui-1.10.4.custom.min.css" rel="stylesheet"></link>
<script src="jquery/js/jquery-1.10.2.js"></script>
<script src="jquery/js/jquery-ui-1.10.4.custom.min.js"></script>
<link href="Badges.css" rel="stylesheet"></link>
<script src="admin.js"></script>
<script>
    $(function() {new admin().configure();});
</script>
</head>
<body>

```

```

<!--<a href="logout.php">Logout</a-->
<div id="Lists" class=" flexContainer">
  <div class="fullFlexItem">
    <h1>Administrator Home</h1>
    <h2>Add Badge</h2>
    <form action="admin.php" method="POST">
      <p>Badge Title:</p><input type="text" name="title" />
      <p>Point Value:</p><input type="text" name="points" />
      <p>Category:</p><input type="text" name="ctg" />
      <p>Description:</p><input type="text" name="desc" />
      <br />
      <input id="createBtn" type="submit" value="Create new Badge" />
    </form>
  </div>
  <div class="fullFlexItem">
    <h2>Badge List:</h2>
    <div id="BadgeList"></div>
  </div>
  <div class="fullFlexItem">
    <h2>User List:</h2>
    <div id="UserList"></div>
  </div>
</div>
</body>
</html>

```

Badges.css

```

html, body {
  height: 98%;
  width: 98%;
  background: #E8E8FF;
  font-family: Arial, sans-serif;
  font-size: 16px;
  color: #444;
}

div#titleBar {
  font-family: italic bold "Comic Sans MS", sans-serif;
  font-size: 3em;
  text-align: center;
  height: 1.5em;
  color: #A0A0FF;
}

div#toolBar, .ui-menu {
  background: #C8C8FF;
}

div#mainArea {
  overflow-y: auto;
  width: 100%;
}

```

```
div#statusBar {  
  font-weight: bold;  
  font-size: 1.5em;  
  height: 1.5em;  
  border: 1px solid;  
  margin-bottom: 10px;  
}
```

```
div#playArea {  
  background-color: #EEF;  
  border: 1px solid #BBD;  
  padding: 1em;  
}
```

```
input#nextMove {  
  width: 5em;  
}
```

```
span#timer {  
  float: right;  
}
```

```
.fixedFlexItem {  
  flex: none;  
  -webkit-flex: none;  
  -ms-flex: none;  
}
```

```
.fullFlexItem {  
  flex: 1 1 auto;  
  -webkit-flex: 1 1 auto;  
  -ms-flex: 1 1 auto;  
}
```

```
.flexContainer {  
  display: flex;  
  display: -webkit-flex;  
  display: -ms-flexbox;  
  overflow: auto;  
}
```

```
.vertical {  
  flex-direction: column;  
  -webkit-flex-direction: column;  
  -ms-flex-direction: column;  
}
```

```
.space-between {  
  justify-content: space-between;  
  -webkit-justify-content: space-between;
```



```

    -ms-flex-pack: justify;
}

.subPanel {
    padding: 1em;
    overflow: auto;
}

.panelHeader {
    font-size: 2em;
    padding-bottom: .4em;
}

.historyEntry {
    height: 1.5em;
    width: 25em;
    margin: 0px;
    border: 1px solid #AAD;
    clear: both;
    background: #DDF;
}

.exact {
    color: #2D2;
    margin-left: 1em;
}

.inexact {
    color: #22D;
    margin-left: 1em;
}

.dialogTemplate {
    display: none;
    border: 1px solid;
    background: EEF;
}

.historyExpansion {
    width: auto;
    padding-left: 1em;
    color: #666;
}

```

Complete.php

```

<?php
$username = "root";
$password = "root";
$hostname = "localhost";

```

```

$dbhandle = mysql_connect($hostname, $username, $password)
or die("Could not connect to database");

$selected = mysql_select_db("badgedb", $dbhandle);

if(!isset($_COOKIE['loggedin'])){
    header("location:index.php");
}
session_start();
$loggedin = $_SESSION['loggedin'];

if (isset($_POST["content_txt"]) && strlen($_POST["content_txt"]) > 0) {
    $title = $_POST["content_txt"];

    $query = "INSERT INTO completed_badges (username, badgetitle)
        VALUES ('$loggedin', '$title')";

    $result = mysql_query($query);

    mysql_close();

    echo $title;
    echo $loggedin;
}
?>

```

delete_badge.php

```

<?php
$username = "root";
$password = "root";
$hostname = "localhost";

$dbhandle = mysql_connect($hostname, $username, $password)
or die("Could not connect to database");

$selected = mysql_select_db("badgedb", $dbhandle);

if(!isset($_COOKIE['loggedin'])){
    header("location:index.php");
}
session_start();
$loggedin = $_SESSION['loggedin'];

if (isset($_POST["content_txt"]) && strlen($_POST["content_txt"]) > 0) {
    $title = $_POST["content_txt"];

    $query = "DELETE FROM completed_badges WHERE
        badgetitle = '$title'";

    $result = mysql_query($query);

```

```

$query = "DELETE FROM badges WHERE
        title = '$title'";

$result = mysql_query($query);

mysql_close();

echo $title;
echo $loggedin;
}
?>

```

delete_user.php

```

<?php
$username = "root";
$password = "root";
$hostname = "localhost";

$dbhandle = mysql_connect($hostname, $username, $password)
or die("Could not connect to database");

$selectd = mysql_select_db("badgedb", $dbhandle);

if(!isset($_COOKIE['loggedin'])){
    header("location:index.php");
}
session_start();
$loggedin = $_SESSION['loggedin'];

if (isset($_POST["content_txt"]) && strlen($_POST["content_txt"]) > 0) {
    $username = $_POST["content_txt"];

    $query = "DELETE FROM completed_badges WHERE
            username = '$username'";

    $result = mysql_query($query);

    $query = "DELETE FROM users WHERE
            username = '$username'";

    $result = mysql_query($query);

    mysql_close();

    echo $username;
    echo $loggedin;
}
?>

```

Index.php

```

<!doctype html>

```

```

<html lang="us">
<head>
<title>CPE Badges</title>

<link href="jquery/css/ui-lightness/jquery-ui-1.10.4.custom.min.css" rel="stylesheet"></link>
<script src="jquery/js/jquery-1.10.2.js"></script>
<script src="jquery/js/jquery-ui-1.10.4.custom.min.js"></script>
<link href="Badges.css" rel="stylesheet"></link>
<script src="LoginDlg.js"></script>
<script>
    $(function() {new LoginDlg().run();});
</script>
</head>
<body class="flexContainer vertical">

</body>
</html>

<div id="loginDlg" title="Cal Poly CPE Badge Service" class="dialogTemplate flexContainer vertical space-between">
<div class="fixedFlexItem">
<form action="login.php" method="POST">
<label for="user">Username:</label>
<input id="user" type="text" name="user" />
<label for="password">Password:</label>
<input id="pass" type="password" name="pass" />
<br />
<button id="loginBtn" class="fixedFlexItem">Login</button>
</form>
</div>
<ul id="errorList" class="fixedFlexItem"></ul>
</div>

```

login.php

```

<?php
$username = "root";
$password = "root";
$hostname = "localhost";

$dbhandle = mysql_connect($hostname, $username, $password)
or die("Could not connect to database");

$selectd = mysql_select_db("badgedb", $dbhandle);

if(isset($_POST['user']) && isset($_POST['pass'])) {
    $myusername = $_POST['user'];
    $mypassword = $_POST['pass'];

    $myusername = stripslashes($myusername);
    $mypassword = stripslashes($mypassword);

    $query = "SELECT * FROM users WHERE Username='$myusername'
and Password='$mypassword'";

    $result = mysql_query($query);
    $count = mysql_num_rows($result);

```

```

mysql_close();

if ($myusername == "admin" && $mypassword == "cpadmin") {
    $seconds = 500 + time();
    setcookie(adminloggedin, date("F jS - g:i a"), $seconds);
    header("location:admin.php");
}
else if ($count==1) {
    session_start();
    $_SESSION['loggedin'] = $myusername;

    $seconds = 500 + time();
    setcookie(loggedin, date("F jS - g:i a"), $seconds);
    header("location:user_home.php");
}else{
    echo 'Incorrect Username or Password';
}
}
?>

```

LoginDlg.js

```

var LoginDlg = function() {
    this.dlg = $('#loginDlg');
    this.user = $('#loginDlg #user');
    this.pass = $('#loginDlg #pass');
    this.loginBtn = $('#loginDlg #loginBtn');
}

LoginDlg.prototype.run = function() {
    var self = this;

    this.loginBtn.hide();
    this.dlg.dialog({
        width: "350px",
        modal: false,
        buttons: [
            //self.loginBtn.appendTo(this);
            {text: "Login", click: function() {self.login()}},
            {text: "New User", click: function() {self.newUser()}}
        ]
    });
}

LoginDlg.prototype.login = function() {
    this.loginBtn.click();
}

LoginDlg.prototype.newUser = function() {
    window.location.href = "new_user.php";
}

```

Logout.php

```
<?
```

```

$seconds = -10 + time();
setcookie(loggedin, date("F jS - g:i a"), $seconds);
header("location:index.php");
?>

```

new_user.php

```

<?php
$username = "root";
$password = "root";
$hostname = "localhost";

$dbhandle = mysql_connect($hostname, $username, $password)
or die("Could not connect to database");

$selectd = mysql_select_db("badgedb", $dbhandle);

if (isset($_POST['user']) && isset($_POST['pass'])
&& isset($_POST['firstName']) && isset($_POST['lastName'])) {

    $user = $_POST['user'];
    $pass = $_POST['pass'];
    $firstName = $_POST['firstName'];
    $lastName = $_POST['lastName'];

    $query = mysql_query("SELECT * FROM users WHERE Username='$user'");

    if (mysql_num_rows($query) > 0 ) { //check if there is already an entry for that username
        echo "Username already exists!";
    }
    else {
        mysql_query("INSERT INTO users (Username, Password, firstName, lastName)
VALUES ('$user', '$pass', '$firstName', '$lastName')");
        header("location:index.php");
    }
}
mysql_close();
?>

```

```

<html>
<body>
    <h1>Signup!</h1>
    <form action="new_user.php" method="POST">
        <p>Username:</p><input type="text" name="user" />
        <p>Password:</p><input type="password" name="pass" />
        <p>First Name:</p><input type="text" name="firstName" />
        <p>Last Name:</p><input type="text" name="lastName" />
        <br />
        <input type="submit" value="Signup!" />
    </form>
</body>
</html>

```

user_home.js

```

var user_home = function() {
    ;
}

user_home.prototype.configure = function() {
    var self = this;
    var uBadgeList = $('#uBadgeList');
    var cBadgeList = $('#cBadgeList');
    var scriptString = 'this string';
    var myData;

    $('#uBadgeList').accordion({collapsible: true});
    for (var i = 0; $('#utitleEntry' + i).length; i++) {
        uBadgeList.append($('#utitleEntry' + i))
            .append($('#udivEntry' + i))
            .append($('#upointsEntry' + i))
            .append($('#ucategoryEntry' + i))
            .append($('#udescriptionEntry' + i))
            .append($('#<button id="completeBtn' + i + '" title="" + $('#utitleEntry' + i).text() + "">Complete Badge</button>')));

        $('#completeBtn' + i).button().on('click', function(e) {
            myData = 'content_txt=' + $(this).attr('title');
            var request = $.ajax({
                url: "complete.php",
                type: "POST",
                data: myData,
                dataType: "html"
            });

            request.done(function(msg) {
                //console.log(msg);
                location.reload();
            });

            request.fail(function(jqXHR, textStatus) {
                alert( "Request failed: " + textStatus );
            });
        });
        console.log($('#titleEntry' + i).text());
    }
    $('#cBadgeList').accordion({collapsible: true});
    for (var i = 0; $('#ctitleEntry' + i).length; i++) {
        cBadgeList.append($('#ctitleEntry' + i))
            .append($('#cdivEntry' + i))
            .append($('#cpointsEntry' + i))
            .append($('#ccategoryEntry' + i))
            .append($('#cdescriptionEntry' + i));
    }
    uBadgeList.accordion("refresh");
    cBadgeList.accordion("refresh");
}

```

user_home.php

```

<?php
$username = "root";
$password = "root";

```

```

$hostname = "localhost";

if(!isset($_COOKIE['loggedin'])){
    header("location:index.php");
}
session_start();
$loggedin = $_SESSION['loggedin'];
print("<h1>Welcome $loggedin</h1>");

$dbhandle = mysql_connect($hostname, $username, $password)
or die("Could not connect to database");
$selected = mysql_select_db("badgedb", $dbhandle);

$query = mysql_query("select badges.title, badges.points, badges.ctg, badges.description
from badges
where badges.title
not in( select b.title
from badges b, completed_badges cb
where b.title = cb.badgetitle AND cb.username = '$loggedin')
group by badges.title");

for ($i = 0; $row = mysql_fetch_array($query, MYSQL_BOTH); $i++) {
    echo "<h3 class='subListEntry' id='utitleEntry$i'>$row[0]</h3>";
    echo "<div class='subListEntryExpansion' id='udivEntry$i'></div>";
    echo "<li id='upointsEntry$i'>Points: $row[1]</li>";
    echo "<li id='ucategoryEntry$i'>Category: $row[2]</li>";
    echo "<li id='udescriptionEntry$i'>Description: $row[3]</li>";
}

$query = mysql_query("select b.title, b.points, b.ctg, b.description
from badges b, completed_badges cb
where b.title = cb.badgetitle AND cb.username = '$loggedin'
group by b.title");

for ($i = 0; $row = mysql_fetch_array($query, MYSQL_BOTH); $i++) {
    echo "<h3 class='subListEntry' id='ctitleEntry$i'>$row[0]</h3>";
    echo "<div class='subListEntryExpansion' id='cdivEntry$i'></div>";
    echo "<li id='cpointsEntry$i'>Points: $row[1]</li>";
    echo "<li id='ccategoryEntry$i'>Category: $row[2]</li>";
    echo "<li id='cdescriptionEntry$i'>Description: $row[3]</li>";
}
?>

<html>
<head>
<title>CPE Badges</title>

<link href="jquery/css/ui-lightness/jquery-ui-1.10.4.custom.min.css" rel="stylesheet"></link>
<script src="jquery/js/jquery-1.10.2.js"></script>
<script src="jquery/js/jquery-ui-1.10.4.custom.min.js"></script>
<link href="Badges.css" rel="stylesheet"></link>
<script src="user_home.js"></script>
<script>
    $(function() {new user_home().configure();});
</script>

```



```
</head>
<body class="flexContainer vertical">
  <div class="fullFlexItem">
    <h2>Unfinished Badges:</h2>
    <div id="uBadgeList"></div>
  </div>
  <div class="fullFlexItem">
    <h2>Completed Badges:</h2>
    <div id="cBadgeList"></div>
  </div>
</body>
</html>
```